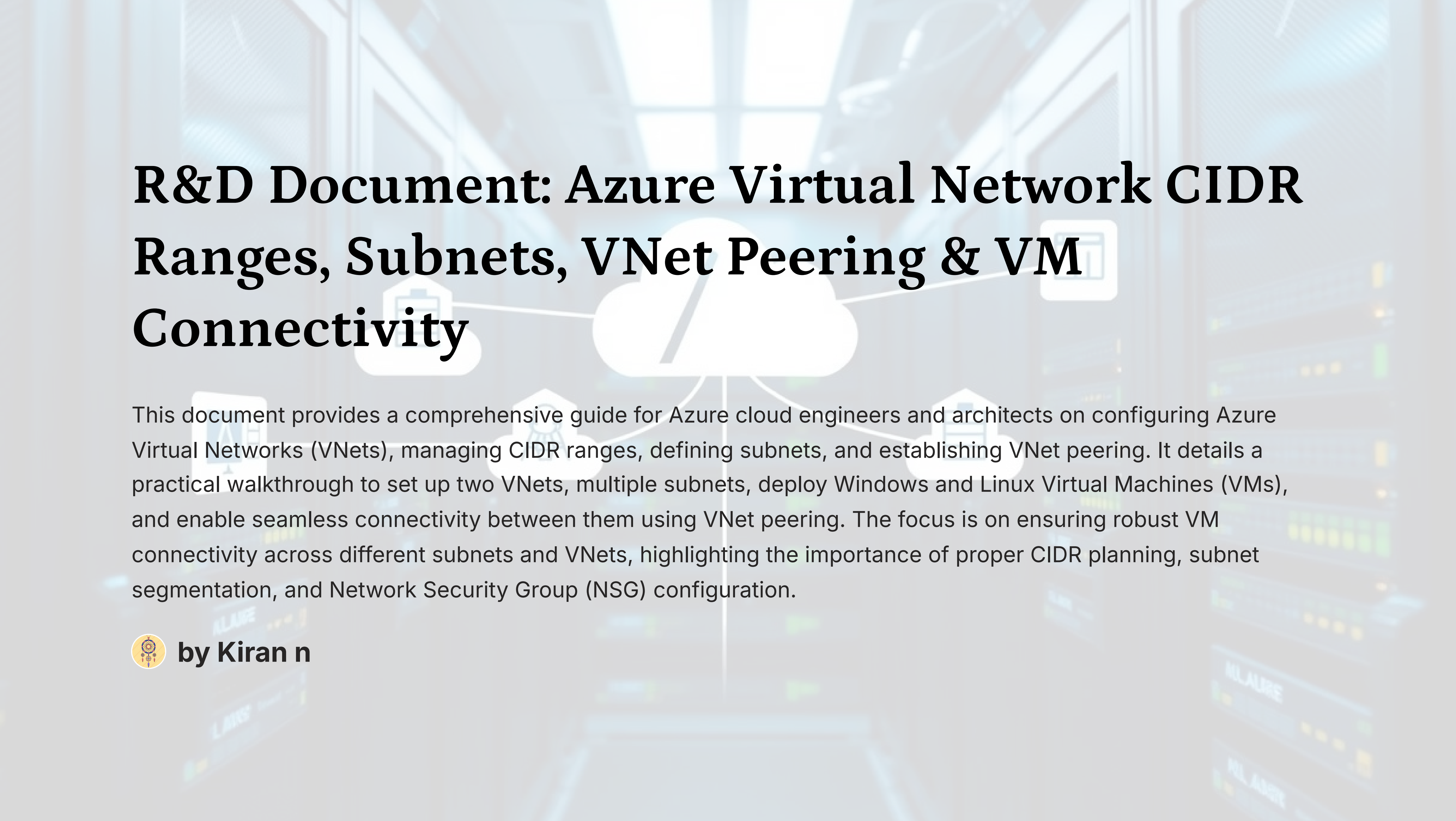


R&D Document: Azure Virtual Network CIDR Ranges, Subnets, VNet Peering & VM Connectivity

This document provides a comprehensive guide for Azure cloud engineers and architects on configuring Azure Virtual Networks (VNets), managing CIDR ranges, defining subnets, and establishing VNet peering. It details a practical walkthrough to set up two VNets, multiple subnets, deploy Windows and Linux Virtual Machines (VMs), and enable seamless connectivity between them using VNet peering. The focus is on ensuring robust VM connectivity across different subnets and VNets, highlighting the importance of proper CIDR planning, subnet segmentation, and Network Security Group (NSG) configuration.

by Kiran n

2. Key Concepts

CIDR Ranges

CIDR (Classless Inter-Domain Routing) is a method for allocating IP addresses and routing IP packets. It extends the original IP addressing system by allowing more flexible specifications of IP addresses and routing tables. In Azure, CIDR notation (e.g.,

```
10.0.0.0/16
```

) defines the address space for Virtual Networks and their subnets. It's crucial that VNet CIDR ranges do not overlap to ensure successful peering and routing. A $/16$ prefix provides a large address space, suitable for a VNet that will contain many subnets, while a $/24$ prefix is common for individual subnets, providing 256 addresses, of which 5 are reserved by Azure.

Azure Virtual Network (VNet)

An Azure VNet is a fundamental building block for your private network in the cloud. It enables many types of Azure resources, such as Azure Virtual Machines (VMs), to securely communicate with each other, the internet, and on-premises networks. A VNet is logically isolated to your Azure subscription, providing a private and secure environment. Within a VNet, you can define multiple subnets, allowing for further segmentation and organization of your resources.

Subnet

Subnets are logical subdivisions within a VNet, allowing you to segment the VNet's address space into smaller, manageable ranges. This segmentation is critical for organizing resources, applying granular security policies (via Network Security Groups), and optimizing network performance. For example, you might create separate subnets for web servers, application servers, and databases, each with its own specific security rules. Each subnet must have a unique CIDR range that falls within the VNet's CIDR range and does not overlap with other subnets in the same VNet.

VNet Peering

VNet peering seamlessly connects two Azure Virtual Networks, enabling traffic between them to flow privately through the Microsoft backbone network. This means that resources in one VNet can communicate with resources in the peered VNet as if they were within the same network, using private IP addresses.

> ### Types of VNet Peering:
>
> - **Intra-Region Peering:** Connects VNets located within the same Azure region. This type of peering offers low-latency and high-bandwidth connections, ideal for applications requiring fast communication between different components deployed in separate VNets in the same region.
> - **Global Peering:** Connects VNets across different Azure regions. This is essential for geographically dispersed applications, disaster recovery strategies, or when you need to connect resources across different Azure geographies. While latency will be higher than intra-region peering, traffic still travels over the private Microsoft backbone, providing security and reliability.

VNet peering is non-transitive. If VNetA is peered with VNetB, and VNetB is peered with VNetC, VNetA cannot directly communicate with VNetC unless a direct peering is established between VNetA and VNetC.

3. Prerequisites

Before proceeding with the implementation steps, ensure that the following prerequisites are met. Adhering to these foundational requirements will streamline the setup process and prevent common configuration errors.

Active Azure Subscription

You must have an active Azure subscription with sufficient permissions to create and manage resource groups, virtual networks, subnets, virtual machines, and network security groups. Ensure your role has Contributor or Owner privileges for the relevant scope.

Resource Group Creation

Two separate Azure Resource Groups are required to logically organize the resources for each Virtual Network. This separation aids in management, billing, and access control.

- **RG-VNet1:** For resources associated with VNet1.
- **RG-VNet2:** For resources associated with VNet2.

These can be created via the Azure Portal, Azure CLI, or Azure PowerShell.

VNet CIDR Planning

Careful planning of VNet and subnet CIDR ranges is essential to avoid IP address overlaps and ensure seamless communication, especially when VNet peering is involved. The following CIDR plan will be used:

- **VNet1:** 10.0.0.0/16
  - **Subnet-Win:** 10.0.1.0/24
  - **Subnet-Linux:** 10.0.2.0/24
- **VNet2:** 10.1.0.0/16
  - **Subnet-Other:** 10.1.1.0/24

This plan ensures that VNet1 and VNet2 have non-overlapping address spaces, which is a fundamental requirement for successful VNet peering.

By establishing these prerequisites, you create a solid foundation for the network configuration, minimizing potential issues during the implementation phase.

4. Step-by-Step Implementation

This section outlines the detailed steps to configure the Azure networking components and virtual machines according to the defined architecture.

Step 1: Create Resource Groups

Navigate to the Azure Portal, search for "Resource Groups", and click "+ Create".

- Create **RG-VNet1** in your preferred region.
- Create **RG-VNet2** in the same region.

Using separate resource groups helps manage permissions and resource lifecycles independently.

Step 2: Create VNets and Subnets

Go to "Virtual Networks" in the Azure Portal and click "+ Create".

- **Create VNet1:**
  - **Name:** VNet1
  - **Resource Group:** RG-VNet1
  - **Address space:** 10.0.0.0/16
  - **Subnets:** Add Subnet-Win (10.0.1.0/24) and Subnet-Linux (10.0.2.0/24)
- **Create VNet2:**
  - **Name:** VNet2
  - **Resource Group:** RG-VNet2
  - **Address space:** 10.1.0.0/16
  - **Subnets:** Add Subnet-Other (10.1.1.0/24)

Step 3: Launch Virtual Machines

Go to "Virtual Machines" and click "+ Create".

- **WinVM:**
  - **Resource Group:** RG-VNet1
  - **Name:** WinVM
  - **Image:** Windows Server (e.g., 2019 Datacenter)
  - **Virtual network:** VNet1
  - **Subnet:** Subnet-Win
  - **Public inbound ports:** Allow selected ports.
  - **Inbound port rules:** RDP (3389) and ICMP enabled.
- **LinuxVM:**
  - **Resource Group:** RG-VNet1
  - **Name:** LinuxVM
  - **Image:** Ubuntu Server (e.g., 20.04 LTS)
  - **Virtual network:** VNet1
  - **Subnet:** Subnet-Linux
  - **Public inbound ports:** Allow selected ports.
  - **Inbound port rules:** SSH (22) and ICMP enabled.

Note: You can configure NSG rules after VM creation, or during the VM creation process by selecting 'Advanced' networking options.

Step 4: Configure NSG (If not done during VM creation)

For each VM's Network Interface Card (NIC), navigate to its associated Network Security Group (NSG). Add an inbound security rule for ICMP:

- **Source:** Any
- **Source port ranges:** *
- **Destination:** Any
- **Destination port ranges:** Any
- **Protocol:** ICMP
- **Action:** Allow
- **Priority:** Choose a priority lower than 65500 (e.g., 100).
- **Name:** AllowICMP

Ensure RDP (for WinVM) and SSH (for LinuxVM) rules are also present.

Step 5: Create VNet Peering

Go to VNet1 in the Azure Portal, select "Peerings" under "Settings", and click "+ Add".

- **Peering link name from VNet1 to VNet2:** VNet1-to-VNet2
- **Virtual network deployment model:** Resource Manager
- **Subscription:** Your Subscription
- **Virtual network:** VNet2
- **Peering link name from VNet2 to VNet1:** VNet2-to-VNet1
- **Allow VNet1 to VNet2 traffic:** Enabled
- **Allow VNet2 to VNet1 traffic:** Enabled
- **Allow forwarded traffic:** Enabled (for future routing needs)
- **Allow gateway transit:** Disabled (unless you have a VPN Gateway in one VNet)

Click 'Add' to create the peering. The status will show 'Connected' once both sides are established.

Step 6: Launch VM in VNet2 and Test Connectivity

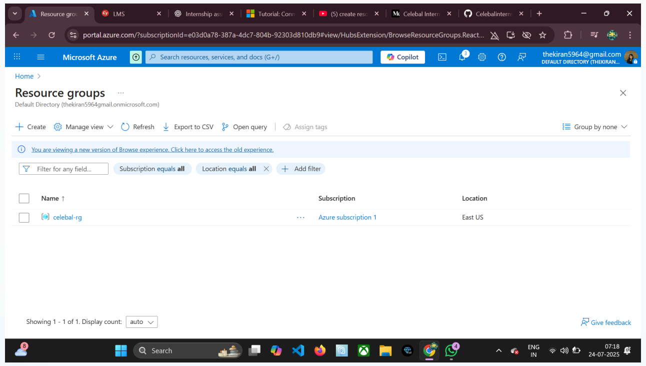
Create a new VM, e.g., VNet2VM, in RG-VNet2, deploying it into VNet2 and Subnet-Other. Ensure ICMP is allowed via its NSG.

Once all VMs are running, retrieve their private IP addresses from the Azure Portal (under "Networking" for each VM).

Log into each VM (RDP for Windows, SSH for Linux) and perform ping tests to the private IP addresses of the other VMs.

6. Screenshots to Capture

Visual documentation is critical for illustrating the successful implementation and for future reference or auditing. The following screenshots should be captured and included in the final R&D document to demonstrate the key configuration steps and validation results.



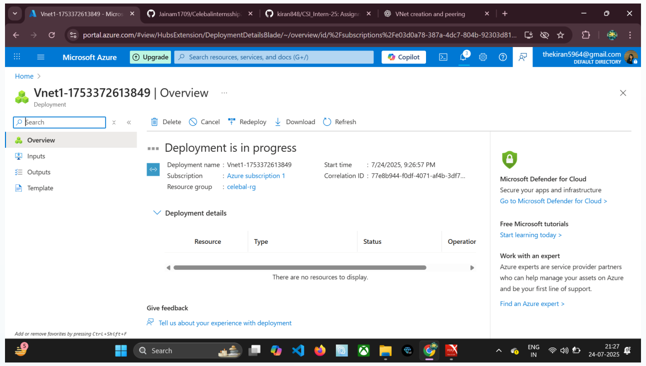
Resource groups

Default Directory (thekiran5964gmail.onmicrosoft.com)

You are viewing a new version of Browse experience. Click here to access the old experience.

Name ↑	Subscription	Location
celebal-rg	Azure subscription 1	East US

Showing 1 - 1 of 1. Display count: auto



portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fsubscriptions%2Fe03d0a78-387a-4dc7-804b-92303d81...
Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
Copilot
thekiran5964@gmail.com
DEFAULT DIRECTORY
Home
Search
Overview
Inputs
Outputs
Template
Delete
Cancel
Redeploy
Download
Refresh
Microsoft Defender for Cloud
Secure your apps and infrastructure
Go to Microsoft Defender for Cloud >
Free Microsoft tutorials
Start learning today >
Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
Find an Azure expert >
Give feedback
Tell us about your experience with deployment

Vnet1-1753372613849 | Overview

Deployment

Deployment is in progress

Deployment name : Vnet1-1753372613849

Subscription : Azure subscription 1

Resource group : celebal-rg

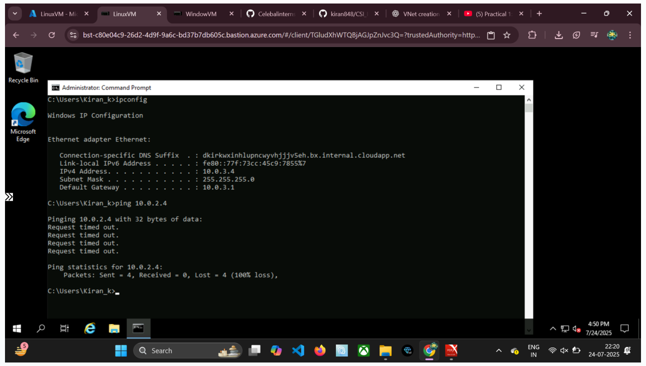
Start time : 7/24/2025, 9:26:57 PM

Correlation ID : 77e8b944-f0df-4071-af4b-3df7...

Deployment details

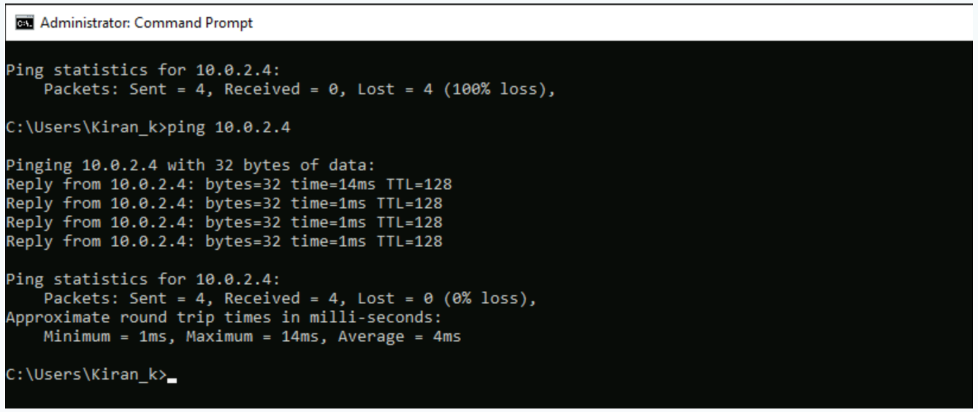
Resource	Type	Status	Operation

There are no resources to display.



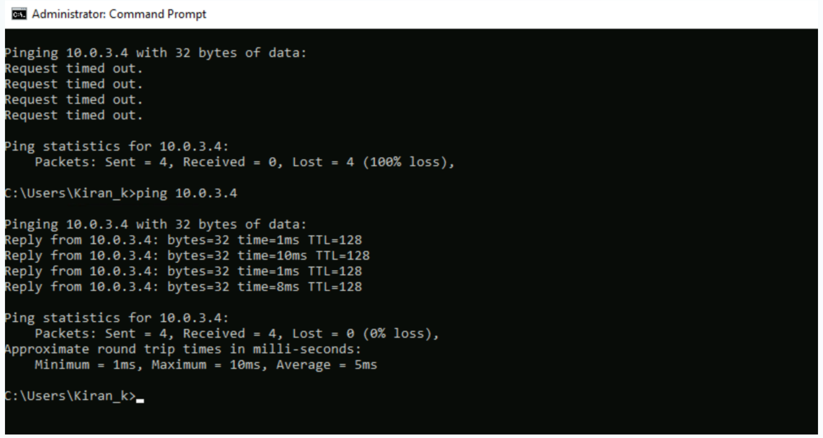
bst-c80e04c9-26d2-4d9f-9a6c-bd37b7db605c.bastion.azure.com/#/client/TGludXhWTQBjAGJpZnJvc3Q=?trustedAuthority=http...
Recycle Bin
Microsoft Edge
Administrator: Command Prompt

```
C:\Users\Kiran_k>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : dkirkwxinhlupncwyvhjjjv5eh.bx.internal.cloudapp.net
   Link-local IPv6 Address . . . . . : fe80::77f:73cc:45c9:7855%7
   IPv4 Address. . . . . . . . . . . : 10.0.3.4
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.3.1

C:\Users\Kiran_k>ping 10.0.2.4

Pinging 10.0.2.4 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.2.4:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Kiran_k>_
```



Administrator: Command Prompt

```
Ping statistics for 10.0.2.4:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Kiran_k>ping 10.0.2.4

Pinging 10.0.2.4 with 32 bytes of data:
Reply from 10.0.2.4: bytes=32 time=14ms TTL=128
Reply from 10.0.2.4: bytes=32 time=1ms TTL=128
Reply from 10.0.2.4: bytes=32 time=1ms TTL=128
Reply from 10.0.2.4: bytes=32 time=1ms TTL=128

Ping statistics for 10.0.2.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 14ms, Average = 4ms

C:\Users\Kiran_k>_
```



Administrator: Command Prompt

```
Pinging 10.0.3.4 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.3.4:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Kiran_k>ping 10.0.3.4

Pinging 10.0.3.4 with 32 bytes of data:
Reply from 10.0.3.4: bytes=32 time=1ms TTL=128
Reply from 10.0.3.4: bytes=32 time=10ms TTL=128
Reply from 10.0.3.4: bytes=32 time=1ms TTL=128
Reply from 10.0.3.4: bytes=32 time=8ms TTL=128

Ping statistics for 10.0.3.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 10ms, Average = 5ms

C:\Users\Kiran_k>_
```

These screenshots provide clear visual evidence of the configuration details and the successful outcomes of the connectivity tests, making the document comprehensive and easy to follow.

7. Additional Notes & 8. Conclusion

7. Additional Notes

Beyond the core steps, consider these important aspects for robust and scalable Azure network deployments:

- **NSG Traffic Rules:** Always ensure Network Security Groups (NSGs) are correctly configured to allow only necessary traffic. Strict NSG rules are fundamental for security posture, acting as a virtual firewall. Remember that NSG rules are evaluated by priority, with lower numbers processed first.
- **Peering Non-Transitivity:** A critical concept in VNet peering is its non-transitive nature. If VNet A is peered with VNet B, and VNet B is peered with VNet C, VNet A and VNet C cannot communicate directly unless a specific peering relationship is established between A and C. This requires careful planning for hub-and-spoke topologies or complex network designs.
- **Global Peering for Different Regions:** When connecting VNets across distinct Azure regions, Global VNet Peering is the required solution. This facilitates cross-region disaster recovery, distributed application deployments, and unified global network architectures while leveraging Microsoft's private backbone infrastructure.
- **Azure Reserved IPs:** Remember that Azure reserves the first four and the last IP addresses within each subnet for internal use. For example, in a `/24` subnet, `.0`, `.1`, `.2`, `.3`, and `.255` are reserved and cannot be assigned to resources. This means a `/24` subnet effectively provides `251` usable IP addresses.

8. Conclusion

This R&D document has meticulously detailed the process of establishing a foundational network infrastructure in Azure, encompassing Virtual Network and subnet creation, strategic CIDR range allocation, deployment of diverse Virtual Machines (Windows and Linux), and the critical configuration of VNet peering.

Through precise planning of non-overlapping CIDR ranges and careful application of Network Security Group (NSG) rules, we successfully demonstrated seamless VM interconnectivity both within the same VNet (across different subnets) and across separate, peered VNets. This validates the effectiveness of Azure's native networking capabilities for creating secure, isolated, yet interconnected cloud environments.

The comprehensive step-by-step guide, combined with the emphasis on testing and troubleshooting, provides a robust framework for Azure cloud engineers and architects to design and implement resilient and scalable network topologies for their workloads.

2. Key Concepts (Continued)

2.1 Understanding CIDR in Azure Networking

CIDR (Classless Inter-Domain Routing) is fundamental to efficient IP address management within Azure virtual networks. It allows for flexible network design by enabling variable-length subnet masking, moving beyond the traditional class-based IP addressing. Each Azure VNet and its subnets are defined by a CIDR block (e.g., `10.0.0.0/16`). The `/16` suffix indicates that the first 16 bits of the IP address define the network portion, leaving the remaining 16 bits for host addresses. This translates to $2^{16} = 65,536$ potential IP addresses within the VNet.

A critical constraint in Azure networking is that CIDR blocks of peered VNets must not overlap. If overlapping CIDR ranges exist, the peering connection will fail, preventing private communication between resources in those VNets. This emphasizes the need for meticulous IP address planning before deployment.

For example, a VNet with a `10.0.0.0/16` address space can be subdivided into smaller subnets. A common subnet size is `/24` (e.g., `10.0.1.0/24`, `10.0.2.0/24`). Each `/24` subnet provides $2^8 = 256$ IP addresses. However, Azure reserves five IP addresses within each subnet for internal use (the first three for internal routing and DNS, and the last two for broadcast and network address). Therefore, a `/24` subnet provides `251` usable IP addresses for your resources.

2.2 Role of Subnets in Azure Architecture

Subnets serve as logical partitions within a Virtual Network, providing granularity for organizing and securing network resources. Their primary roles include:

- **Resource Isolation:** Subnets allow for the segregation of different application tiers (e.g., web servers, application servers, database servers) into distinct network segments. This isolation limits the blast radius in case of a security breach.
- **Security Policy Enforcement:** Network Security Groups (NSGs) can be associated directly with subnets. This means you can apply specific inbound and outbound rules to all resources within a particular subnet, enforcing granular traffic control based on the workload's security requirements. For instance, a database subnet might only allow inbound traffic from the application subnet.
- **Route Table Association:** Custom route tables can be associated with subnets to control traffic flow more precisely, overriding Azure's default routing behavior. This is useful for scenarios involving network virtual appliances (NVAs) or forced tunneling to on-premises networks.
- **Delegated Subnets:** Certain Azure services, like Azure App Service Environment or Azure Container Instances, require delegated subnets. These subnets are exclusively used by the service, providing network integration and sometimes custom NSG configurations.

2.3 Deep Dive into VNet Peering Types

VNet peering facilitates direct, private IP connectivity between Virtual Networks. Understanding the nuances of each peering type is crucial for designing resilient and globally distributed Azure architectures:

Intra-Region Peering	Connects VNets within the same Azure region. Traffic remains on the Microsoft backbone, offering low-latency and high-throughput connections. Ideal for splitting large applications into multiple VNets for better management, isolating environments (Dev/Test/Prod), or extending network reach within a single region without complex routing.
Global VNet Peering	Connects VNets across different Azure regions. Traffic traverses the Microsoft global backbone network, ensuring privacy and reliability. Essential for geo-redundant applications, disaster recovery scenarios, global load balancing, or providing services to users in different geographical locations while maintaining a unified network experience. Latency is higher than intra-region, but the traffic does not leave the Microsoft network.
Cross-Subscription Peering	Allows peering between VNets that belong to different Azure subscriptions but are within the same Azure AD tenant. This is common in large enterprises with multiple subscriptions for different departments or projects, enabling resource sharing and communication across organizational boundaries while maintaining central governance. This can be either Intra-Region or Global.

A key characteristic of all VNet peering types is that they are **non-transitive**. This means if VNet A is peered with VNet B, and VNet B is peered with VNet C, VNet A cannot communicate directly with VNet C unless a direct peering is established between VNet A and VNet C. This design choice ensures network security and explicit control over traffic paths.

4. Step-by-Step Implementation (Continued)

4.1 Designing the Network Topology

For this practical use case, the network topology is designed to illustrate both intra-VNet and inter-VNet communication. We've established two distinct Virtual Networks:

- **VNet1 (10.0.0.0/16):** This VNet will host two subnets, Subnet-Win (10.0.1.0/24) and Subnet-Linux (10.0.2.0/24). A Windows Virtual Machine (WinVM) will be deployed in Subnet-Win, and a Linux Virtual Machine (LinuxVM) will reside in Subnet-Linux. This setup allows us to demonstrate connectivity between different subnets within the same VNet.
- **VNet2 (10.1.0.0/16):** This VNet contains a single subnet, Subnet-Other (10.1.1.0/24), which will host a third Virtual Machine (VNet2VM). The non-overlapping CIDR ranges for VNet1 and VNet2 are crucial for establishing successful VNet peering.

The primary objective is to enable seamless communication between all three VMs, demonstrating how VNet peering bridges the communication gap between VNet1 and VNet2.

4.2 Why Create Separate Subnets for VMs?

The decision to segment VMs into different subnets, even within the same VNet, is based on best practices for network design and security:

- **Enhanced Security:** By associating Network Security Groups (NSGs) with specific subnets, you can implement fine-grained security policies. For instance, you might allow RDP only to the Subnet-Win and SSH only to Subnet-Linux from specific source IPs, while restricting all other inbound traffic.
- **Traffic Monitoring and Control:** Subnet-level segmentation facilitates easier monitoring of traffic flows using tools like Azure Network Watcher and NSG Flow Logs. It also enables the application of custom route tables to direct traffic from a subnet through a specific network virtual appliance (NVA) for inspection or routing.
- **Mimicking Production Architectures:** In real-world enterprise deployments, applications are often deployed across multiple tiers (e.g., web, application, database). Each tier typically resides in its own subnet to ensure logical separation, controlled communication, and scalable resource allocation. This lab setup replicates such a scenario.
- **Scalability and Management:** Smaller, dedicated subnets are easier to manage and scale. Adding or removing resources or applying changes to network configurations is less disruptive when resources are logically grouped.

4.3 Security Group Configuration

Azure Network Security Groups (NSGs) act as a virtual firewall for your VMs, controlling inbound and outbound network traffic. By default, Azure NSGs are highly restrictive, blocking most inbound connections to ensure security. To enable the required communication for this lab:

- **ICMP:** To facilitate basic connectivity testing using the ping command, an inbound NSG rule explicitly allowing ICMP traffic must be added to the NSGs associated with the NICs of all three VMs. This rule should permit ICMP from any source to any destination within the VNet.
- **RDP (Port 3389):** For remote administration of the Windows VM (WinVM), an inbound rule allowing RDP traffic on port 3389 must be configured. For security best practices, it's highly recommended to restrict the source IP range for RDP to only your trusted IP address, rather than allowing from 'Any'.
- **SSH (Port 22):** Similarly, for remote administration of the Linux VM (LinuxVM), an inbound rule allowing SSH traffic on port 22 is required. As with RDP, restrict the source IP range for SSH for enhanced security.

These rules ensure that essential management and testing protocols are permitted while maintaining a secure posture.

4.4 Peering Configuration Explained

When establishing VNet peering, careful consideration of the configuration options is paramount for achieving the desired network behavior:

- **Bidirectional Traffic Flow:** It is imperative to enable traffic flow in both directions (e.g., "Allow VNet1 to VNet2 traffic" and "Allow VNet2 to VNet1 traffic"). Forgetting to enable one direction will result in one-way communication only, leading to connectivity issues. Azure creates two peering objects, one for each VNet, and both must be in a "Connected" state.
- **Allow Forwarded Traffic:** Enabling "Allow forwarded traffic" is important if you have a Network Virtual Appliance (NVA), such as a firewall, in one VNet and want to route traffic from the peered VNet through this NVA. This option allows traffic that is not directly addressed to the VNet itself (i.e., traffic from other peered networks) to be forwarded.
- **Allow Gateway Transit:** This option is used when one VNet has a VPN Gateway (either Site-to-Site or ExpressRoute) and you want to allow other peered VNets to use this gateway to reach on-premises networks. The VNet with the gateway is the "hub," and other VNets are "spokes." Only one side of the peering should be configured for gateway transit; the other side should use "Use remote gateways."

By correctly configuring these peering options, you ensure that traffic flows as intended, supporting complex network architectures and connectivity requirements.

5. Testing and Validation (Continued) & 7. Additional Notes (Continued)

5.1 Verifying Connectivity

After deploying VMs and configuring network settings, verifying connectivity is a crucial step to ensure the network is functioning as intended. The `ping` command is a simple yet effective tool for this purpose. When performing ping tests, observe the following:

- **IP Address Source:** Confirm that the VM receives a private IP address from its assigned subnet's CIDR range (e.g., `10.0.1.x` for `Subnet-Win`).
- **NSG Rule Efficacy:** A successful ping indicates that the inbound ICMP rule on the target VM's NSG is correctly configured and allowing traffic. If ping fails, the NSG is often the first place to check.
- **Peering Validation:** Successful pings between VMs in different VNets (e.g., `WinVM` in `VNet1` to `VNet2VM` in `VNet2`) confirm that the VNet peering is established and traffic is flowing correctly across the Azure backbone.
- **Network Latency:** Observe the ping response times. While intra-VNet pings should be very low latency, inter-region global peering will naturally show higher latency, which is expected.

5.2 Troubleshooting Tips

Network issues can be complex. If connectivity tests fail, consider the following troubleshooting steps:

- **NSG Rules:** Double-check all inbound and outbound NSG rules associated with the VM's network interface and the subnet itself. Ensure that ICMP, RDP, and SSH are explicitly allowed from the correct source IP ranges. Remember that NSG rules are processed in order of priority, and a deny rule with a lower priority number can override an allow rule with a higher priority number.
- **Peering Status:** In the Azure Portal, navigate to the VNet peering section for both VNets. Ensure the "Peering status" is 'Connected' for both sides of the peering link. If it's 'Initiated' or 'Disconnected', the peering is not fully established.
- **VM Firewall:** Even if Azure NSGs allow traffic, the operating system's internal firewall (e.g., Windows Defender Firewall, firewalld/ufw on Linux) might be blocking ICMP or other ports. Temporarily disable the OS firewall for testing, or add specific inbound rules for the required protocols.
- **Correct Private IPs:** Always use the private IP addresses of the target VMs for ping tests within the Azure network. Public IPs are used for external access, but internal Azure network communication relies on private IP addresses.
- **IP Overlaps:** Re-verify that no CIDR ranges between peered VNets or within the same VNet's subnets are overlapping. This is a common cause of peering failures.
- **Azure Network Watcher:** Utilize Network Watcher's IP Flow Verify and Connection Troubleshoot features to diagnose specific connectivity problems. These tools can identify if an NSG rule or a routing issue is blocking traffic.

7.1 Best Practices for Enterprise Deployment

Building on the foundational concepts demonstrated, consider these best practices for designing robust and scalable Azure networks in an enterprise environment:

- **Azure Policy for Naming Conventions:** Implement Azure Policies to enforce consistent naming conventions for all network resources (VNets, subnets, NSGs, VMs, etc.). This improves resource identification, management, and auditing, especially in large environments.
- **Restrict Subnet Access with NSG and ASG:** Beyond basic NSG rules, leverage Application Security Groups (ASGs). ASGs allow you to group VMs based on their application function (e.g., WebServers, AppServers) and then create NSG rules that reference these ASGs, simplifying security management as your application scales. This avoids having to update NSG rules every time a new VM is added.
- **Monitor VNet Traffic with Network Watcher:** Enable Azure Network Watcher and NSG flow logs. This provides detailed information about all traffic flowing through your NSGs, including source/destination IPs, ports, and protocols, which is invaluable for security auditing, compliance, and troubleshooting.
- **Centralized Logging and Analytics:** Integrate NSG flow logs and other network diagnostic data with Azure Log Analytics workspaces. This enables centralized logging, powerful querying, and advanced analytics for network performance, security insights, and anomaly detection.
- **VM Deployment Behind Load Balancer/Application Gateway:** For production workloads, always deploy VMs behind Azure Load Balancer (for TCP/UDP load balancing) or Azure Application Gateway (for web traffic load balancing and WAF capabilities). This provides high availability, scalability, and enhanced security by abstracting VMs from direct internet exposure.
- **Hub-and-Spoke Topology:** For complex organizations, consider a hub-and-spoke network topology using VNet peering. A central "hub" VNet contains shared services (e.g., VPN Gateway, firewalls, DNS), while "spoke" VNets host individual workloads. This centralizes security, routing, and management.
- **UDRs (User Defined Routes):** Implement User Defined Routes on subnets to control traffic flow to specific next hops, such as a Network Virtual Appliance (NVA) for centralized firewalling or to force all outbound internet traffic through an on-premises proxy.